Writing Threaded Apps on Mac OS X

Session 112

Writing Threaded Apps on Mac OS X

Mark Tozer
Hardware Evangelist

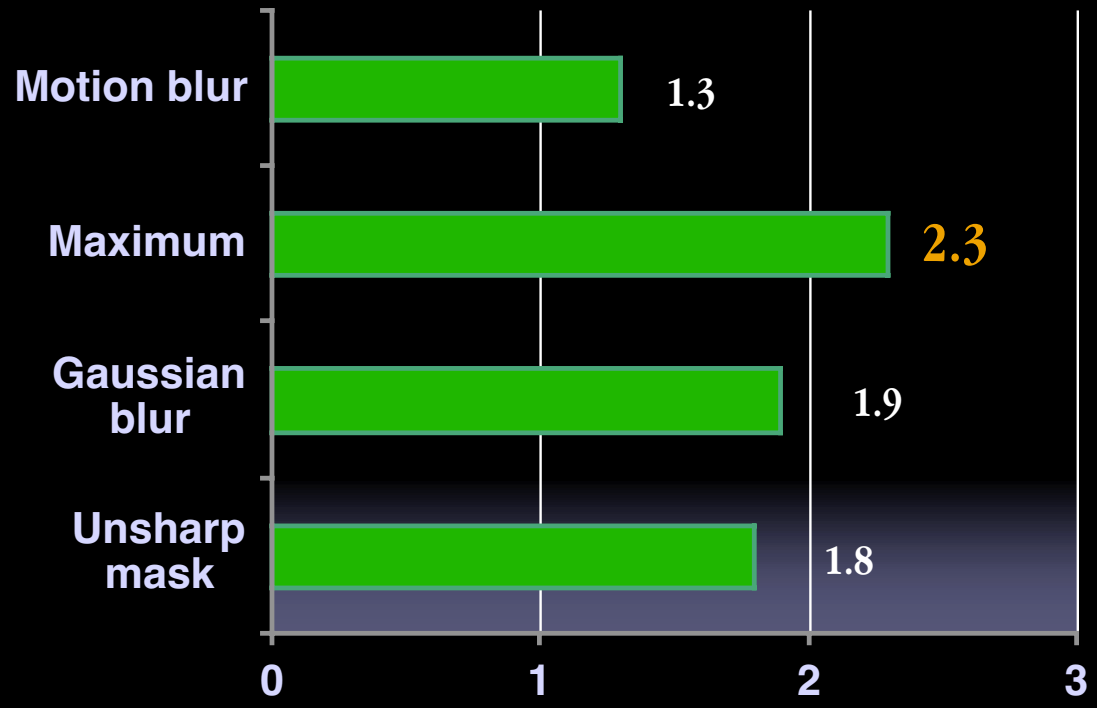
Writing Threaded Apps on Mac OS X

Matt Watson
Core Technologies Engineer

Why Use Threads?

- Customer expectation
- Scalability
- Preemption
- Synchronous requests
- Polling is bad
- Data-driven tasks

MP Scaling


Motion blur
1.3
Maximum
2.3
Gaussian blur
1.9
Unsharp mask
1.8
0
1
2
3

When to Avoid Threads?

- Added complexity
- Added overhead
- Kernel resources
- Other options (timers)
- 100s of threads

Thread Overhead

- Context switch
  - float/vector state
- Memory footprint
  - 512KB default stack
- Creation time

Common API Concepts

- How to use threads
  - Thread management
  - Synchronization primitives
- What can be called
  - Thread-safe services

Internal Implementation

- Threads are the scheduling primitive
- Fully Preemptive
- Priority-based scheduling
- 1–1 model
- Mach threads—look but do not touch

Mach

- Full SMP
- Single kernel for UP and MP
- Lazy floating/vector context save

Mach Scheduler

- OSF scheduling framework
- Global run queue
- Idle processor signal
- Dynamic thread affinity
- Preemption

Pthreads


Carbon
MP API
Cocoa
NS Threads
Java
Java Threads
Darwin
Posix Threads

Pthreads

- Basis for all thread models
- “POSIX-like” implementation
- 1–1 Mach-to-pthread
- Common API (mis)uses
- No system-wide types (yet)

Pthreads (Cont.)

- Synchronization is not cheap
- Do not forget to detach
- Limit your stack
- Check predicates!
- pthread cancel()
    - Deferred good
    - Async bad

Pthreads Details

- pthread_cancel()
  - pthread_testcancel()
  - system-defined cancellation points
  - unblocks pthread_cond_wait()

Pthreads Details (Cont.)

- pthread_atfork()
  - Used by libraries/frameworks
  - Watch out for deadlock!
  - Restricted use

Pthreads Details (Cont.)

- pthread_condattr*()
- pthread_mutexattr*()
  - Mostly for ease of porting
  - System wide not supported
- pthread_attr_set/getstack()
  - SUSv3 addition

Pthreads Details (Cont.)

- pthread_kill()
- pthread_sigmask()
- sigwait()
    - Interrupting threads
    - Porting aid, e.g., Apache 2.0

Pthreads Details (Cont.)

- pthread_rwlock*()
  - Prefers writers
  - Now mandatory in SUSv3
- New mutex types
  - PTHREAD_MUTEX_ERRORCHECK
  - PTHREAD_MUTEX_RECURSIVE

Pthreads References

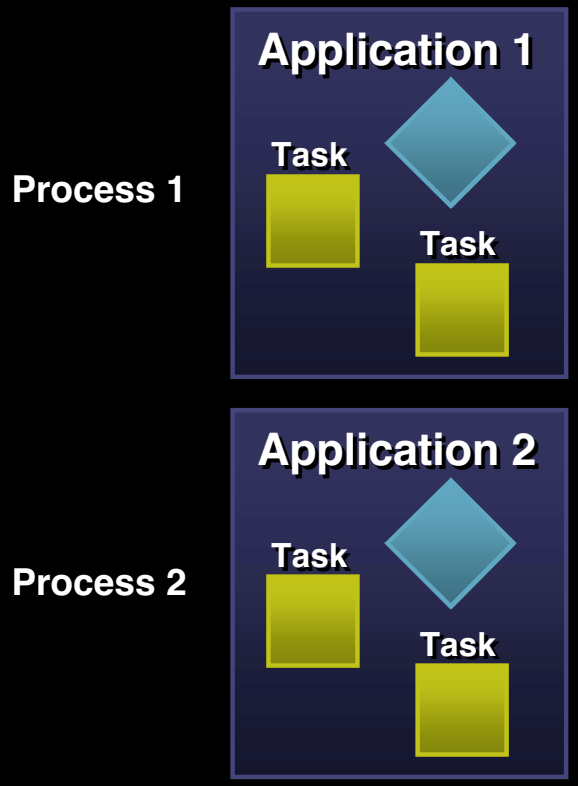
- **man** pages
- Darwin CVS repository
  - Libc
  - xnu

http://www.opengroup.org

news://comp.programming.threads

Carbon MP API

MP Tasks



- Address space
- MP tasks are associated with parent task's process
- Separate address spaces

MPTask API

- Several mechanisms exist to coordinate and synchronize tasks
  - Semaphores
  - Message Queues
  - Event Groups
- Critical regions
  - Can handle recursive entries and multiple entry points

MPTask API (Cont.)

- Atomic inc/dec operations
- APIs on Mac OS X
    - Synchronous File Manager
    - Open Transport
- Thread-safe services

TechNote 2006

Cocoa NSThreads

NSThread

- Simple to use
- Objective-C exceptions
- Exit notification
- Per-thread NSDictionary

NSThread (Cont.)

- AppKit extension
- Separate run loop
    - Need to run explicitly
- Autorelease pool

Future Developments

- Priority inheritance
- System-shared resources
- Performance
- Watch the Darwin repository

Demo

Robert Bowdidge
Developer Tools

Roadmap

102 Performance Optimization With Velocity Engine	Room J **Tue., 9:00am**
905 Apple Performance Tools	Hall 2 **Thurs., 5:00pm**
906 Developing for Performance	Hall 2 **Fri., 9:00am**
407 Java Performance	Room C **Fri., 9:00am**

Who to Contact

Matt Watson
Core Technologies Engineer
mwatson@apple.com

Mark Tozer
Hardware Evangelist
tozer@apple.com

http://developer.apple.com/wwdc2002/urls.html

Documentation

Threading

- Carbon Threads

  Documentation > Carbon > Operating System Services > Multiprocessing

  developer.apple.com/techpubs/macosx/Carbon/oss/MultiPServices/multiprocessingservices.html

- Cocoa Threads

  Documentation > Cocoa > Process Management > Multithreading

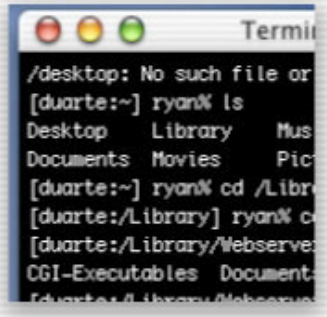
  developer.apple.com/techpubs/macosx/Cocoa/TasksAndConcepts/ProgrammingTopics/Multithreading/index.html

More Documentation

Threading

- Technical Note 2028: Threading Architectures
  developer.apple.com/technotes/tn/tn2028.html
- Technical Note 2006: MP-Safe Routines
  developer.apple.com/technotes/tn/tn2006.html

Q&A



Mark Tozer
Hardware Evangelist
tozer@apple.com

http://developer.apple.com/wwdc2002/urls.html


WWDC2002


WWDC2002


WWDC2002